Import Wizard Version 8.2.1

Copyright 1996-2005 Beside Company Inc.

Quick Start (learn to use Import Wizard in 15 minutes)

Notices to Upgraders:
- Make a backup copy of your model files before using verson 8. Version 8 will open models created by previous versions, but previous versions might not open version 8 models correctly.
- The Access/Excel VBA interface changed, see Visual Basic Functions
- The impwiz.exe command line option have been extended, see Command Line Options

Introduction

Import Wizard enables you to easily import text based report files into databases or spreadsheets. The software goes beyond the standard text import functionality by allowing imports with multiple lines per record and importing headers and footers. The software will prove to be very useful to import information from legacy systems, mainframe, non-windows based, or secured systems. The imported information can be manipulated as a table in the database into which it was imported. How does it work? Ninety-nine percent of software systems have a provision to print reports to a printer spool file instead of printing directly to a printer. This software uses these printer spool files to create tables. It accomplishes this by using an import model. The software includes a wizard to define these models within minutes. This software will save you a lot of headaches trying to figure out a way to transfer data between systems.

Import Wizard is available as executable file to export to delimited text files or ODBC databases and as menu add-in for Access and Excel 97, 2000, XP(2002), and 2003. Import Wizard runs on Windows 95, 98, ME, NT4, 2000 and XP operating systems.

For developers the Import Wizard Software Developer Kit is available. With the SDK you can embed Import Wizard into your own application. Please visit www.beside.com for more information, licensing and downloads.

Software updates for the same main version number as you purchased (8.x.x) are free of charge and can be downloaded from our website: www.beside.com

General Information

Installation

Uninstall
Ordering Software, Registration
Copyrights, Warranty, License Agreement

User's Manual

Quick Start and Import Example 1: Simple Import
Import Example 2: Header Imports
Other Import Examples
Main Import Wizard Window
Config Dialog
Select Import Type
Importing Fixed Position Files
Importing Delimited Files
Importing HTML Files
Importing Excel Files

Reference Manual

What's New
Formulas
Regular Expressions
Visual Basic Functions (Add-in Versions Only)
Connect Strings (Pro Version Only)
Command Line Options (Pro Version Only)
Convert To and Conversion Method Field Properties
Date and Numeric Format Strings (Pro Version Only)
Import Error and Warning Messages
Marker Pattern Definitions
Delimiter Characters Definitions
Import Algorithm and Limitations
International Versions

Installation

1) It is optional to uninstall previous versions of Import Wizard before installing an update. See Uninstall for instructions. To update a previous version 8 to the current version 8 simply download the latest version from www.beside.com and run the setup program without unstalling the previous version 8, this will keep your settings including registration key.

2) Run the Setup.exe program.

3) Follow the prompts to install the software in a directory. Select the components you want to install. If at a later time you want to add components, simply run setup again.

4) To start the program: use your Start menu: *Start->Programs->Import Wizard*. The Access and Excel add-in versions can be started from within Access/Excel: open any database or spreadsheet and select menu *Tools->Import Wizard*.

Note: Import Wizard requires the Microsoft Visual Basic 6.0 Runtime module. This module is pre-installed on Windows 98 or later. If you get "Object variable or With block variable not set" errors, you are probably missing this module. Download (from www.beside.com) and install this module. You can install the module before or after you installed Import Wizard.

Note for Import Wizard Pro only: Import Wizard Pro uses Microsoft Data Access Components (MDAC) in combination with the Jet Database Engine version 4.0. If you have fully installed Office 2000 (including Access 2000) on your system, then you should have these components installed already. If not, or if you want to update MDAC to the latest version, you can download them directly from Microsoft:
MDAC information: http://www.microsoft.com/data
MDAC download: http://msdn.microsoft.com/data/downloads/updates/default.aspx#MDAC
Jet 4.0: http://support.microsoft.com/support/kb/articles/q239/1/14.asp

Uninstall

Note: Uninstall removes the registration code from your computer, make sure that you have a copy of the registration code in order to re-install the software at a later time.

To uninstall Import Wizard: Run *Start->Programs->Import Wizard->Uninstall* from the Windows Start menu. Alternatively you can select menu *Start->Settings->Control Panel->Add/Remove Programs* and remove the "Import Wizard" entry from the list by clicking the "Add/Remove" button.

To uninstall certain components of Import Wizard: first uninstall Import Wizard, then use the installation procedure to install only the desired components.

After running uninstall you can delete the directory where you installed the software to remove files that were created after the software was installed.

Ordering Software, Registration

This is not free software. You may not use this software beyond the initial 30 day evaluation period unless you register the software. The following ordering options are available:

- **Secure online Visa/MasterCard/Amex/Diners transaction via www.beside.com, you will have your registration code within minutes!**
- Check, international money order, wire transfer, or credit card endorsement sent. Please visit www.beside.com.

Upon receipt of your registration fee you will be sent a registration code which can be recorded in your copy of the software. The registration code and name in which you registered need to be entered into the Import Wizard software via the registration screen. To get to the registration screen use Register in the Help menu of main Import Wizard screen. **Once the registration code is entered the software will indicate that the copy is registered and imports of any size are possible. (Only limited by the capabilities of the database system you are using)**

Registered users can obtain, when available, the next version 8.x.x revision of the software free of charge from our web site: www.beside.com. Technical support is provided via email to impwiz@beside.com for the period of one year after the purchase date.

Registration Fees

The registration fee for Import Wizard 8 covers Import Wizard Pro and all add-in versions.

Single User License:US$ 149

For Multi-User, Site, SDK, or Source Code licenses pricing please visit www.beside.com

See Copyrights, Warranty, and License Agreement for a full description of the licensing conditions.

Services

We can help you build a data transformation solution by programming an Import Wizard Model for you. This service costs US$100, and can be ordered via: www.beside.com. After receiving a sample source file and a description of the requirements we usually have a solution ready within 1 business day, please visit www.beside.com for up to date information.

Quick Start, Import Example 1

See also:
Import Example 2: Header Imports
Other Import Examples
The Main Import Wizard Window

To start Import Wizard: use your Start menu: *Start->Programs->Import Wizard*. The Access and Excel Add-in versions are started from within Access/Excel: open any Access database or Excel spreadsheet and select menu *Tools->Import Wizard*.

Import Wizard uses so-called Models that define how a file is imported. These models are stored as regular files, with an ".iwm" extension. First step in importing a file is to define a model for the import. The software includes a wizard that helps you build a model. The main screen of the Import Wizard shows the model parameters, i.e. what file is imported into what table. You can view and modify the model details by pressing the Modify button or menu Model File->Modify.

The following example explain in detail how to use the software. All sample files can be found in the "Samples" subdirectory. In this directory you find the report files to be imported (*.txt), and the finished model files (*.iwm).

Import Example 1: Simple Import

This is an example that illustrates how to import a simple report file with Import Wizard. Printed below is the content of the report file "sample1.txt" that is used in this example:

```
============================================================
Employee Hours Report                              Page:  1
Company: XYZ Company                  Date Printed:02/02/98
============================================================
Employee           Date     Regular Hours/    Other Hours/
                            Overtime              Code
------------------------------------------------------------
Doe, John        01/23/98     23.3                  1.6
                              15.6                   AC
------------------------------------------------------------
Wacks, Gene      02/01/98      1.6                  0.0
                               0.0
------------------------------------------------------------
Peters, Kate     01/30/98     40.3                  3.4
                               1.1                   VA
------------------------------------------------------------
```

From this report file we want to create the following table:

Employee	Date	Regular Hours	Other Hours	Overtime Hours	Code
Doe, John	1/23/99	23.3	1.6	15.6	AC
Wacks, Gene	2/1/99	1.6	0	0	
Peters, Kate	1/30/99	40.3	3.4	1.1	VA

Using standard fixed width text import program you will have a number of difficulties to overcome before you have the desired result. First you will have to get rid of the header information and the dividing lines between the records, then you have to figure out a way to combine two imported lines per record into a single record. Import Wizard handles these tasks for you.

To get a quick demonstration: open the “sample1.iwm” model file (in the “Samples” directory) with the Open button, or from the menu Model File->Open. Then click the Import button, or menu Import->Preview Import. After a short delay you will see the contents of the Sample1 table that you just created. The table contains three records, corresponding with the data found in the report file “sample1.txt”.

Having seen the results, the next thing to do is to actually build the import model yourself. Follow these instructions to build the model for this import:

1.Create a new import model

Press the New button, or select Model File->New from the menu if you are using the standalone version Import Wizard.

2.Select file to import

In the file dialog select the report file “sample1.txt” from the “Samples” directory.

3.Select the type of file to import

Select Fixed, and click the Next> button.

4.The Fixed Model Design dialog

You now see the Fixed Model Design dialog. The dialog appears to be very complex at first sight, but once you understand the basics, this dialog will allow you to define import models conveniently and quickly.

The Fixed Model Design dialog is split in three sub-windows:

File Window The top half of the screen contains the File window which shows the file content. Lines in the File window that match the selected marker are highlighted in yellow. If a field is selected then the fields are also highlighted (in blue) in the File window.

Model Window Bottom left is the Model window which gives an overview of the markers and fields defined in the model. When you select a marker or field in this window the File window highlights the part of the file that matches the selected item and the Properties window will show the properties for the selected item.

Properties Window Bottom right is the Properties window which gives the property details for the selected marker or field in the Model window.

5.Define marker pattern

The first step in defining a model is to define a marker pattern. A marker pattern will mark (match) certain lines in the source file, these marked lines can then be used to import fields. You control which lines are marked by editing the Pattern input box at the top of the File Window.

The pattern works similar as wildcards in filenames. For example if you want to list all the filenames that are two characters long and end with .asc you enter “??.asc”, where the “?” is a wildcard and stands for any single character, and the “.”, “a”, “s”, and “c” have no special meaning and match the same character.

Similar in the Pattern. The following wildcard can be used in the Pattern:

Space matches any character.

? matches any character except space.

@ matches an alpha character (“a” to “z” and “A” to “Z”)

matches a digit character (“0” to “9”)

_ matches a space character (“ “).

Any character that is not a wildcard will match the same character (case sensitive).

In this example we want the marker to match the lines with the employee names, as these lines are the start of the records we want to import. Having determined which lines have to be matched, next task is to find out what differentiates these lines from all other lines in the file.

Looking at the file, you see that the lines we want to match contain a date and that this date does not appear in the other lines, thus making this date an appropriate marker. To create the marker, place the cursor in the Pattern text box at the top of the File Window and enter the following pattern (without the quotes):

" ##/##/##"

This Pattern tells Import Wizard that you are looking for lines that start with 17 characters that can be any character (the 17 spaces), followed by two digits (##), followed by a slash character (/), then two digits (##), a slash (/) and again two digits (the last “##”).

In order for the marker pattern to work properly, the ##/##/## has to be aligned with the dates in the file. This alignment is most easily done by scrolling the file until the a line that needs to be matched is directly below the pattern input box.

As you enter the marker, you will notice that lines the file are highlighted in yellow. If you entered the marker pattern correctly, the 3 lines that contain the employee names light up. If you do not get this result, double check your Pattern, probably you typed a space too few or

too many.

6.Define Fields

You can create new fields by selecting text in File window. Simply move the mouse pointer to the start of the text you want to import, press the left mouse button, drag the mouse to select the text you want to import into this field, then release the mouse button. If you just click a line without selecting at least one character you will be asked if you want to create a marker. Just click No, and try again.

Drag your mouse over an employee name, for example “Doe, John”. Make sure to include enough spaces after the name to correctly import longer names. After you release the mouse button a new field is appended to the Model window, and the File windows highlights the field in blue. Not only the employee name you just selected is highlighted also the other two names in the file. You can now change the Field Properties as desired, in our case change the field name from “Field1” to “Employee”.

Do the same with the "Date", "Regular Hours", "Overtime Hours", "Other Hours", and "Code" fields. If you make an error in dragging your mouse, you can delete the field by pressing the Delete button, and try again. Alternatively you can change the field properties manually.

7.Test your import model

Now that you defined the import model, press the “Import Preview” button to see the results. This will open a popup window, which contains the three records we wanted to import. If the results are different, you have to make the appropriate changes to your model and try again.

8.Finishing up

Once you verified that the model works correctly, press the “OK” button on the Fixed Model Design dialog. This brings you back to the main Import Wizard screen and you see the model properties of the newly created model. You can modify these properties as required, for example you can enter a different output table name. Save the model by pressing the “Save” button or selecting Model -> Save from the menu. Run the import by pressing the “Import” button or selecting Import -> Import from the menu.

Import Example 2: Header Imports

See also:

[Import Example 1: Simple Import]

[Other Import Examples]

[The Main Import Wizard Window]

In this example we will build a model for the "sample2.txt" report file. The file contains this report:

```
============================================================
Product Sales Report                                 Page:  1
Company: XYZ Company          For the month ended:09/30/98
============================================================
Sales Person       Date         Qty        Price     Class
------------------------------------------------------------
            Product: Tape

Doe, John        09/23/98     1,120     11,225.60        AC
                 09/24/98    21,123    233,112.67        VA
                 09/27/98        23         32.22        CS
Wacks, Gene      09/01/98     3,766      3,455.55        AC
Peters, Kate     09/30/98     4,000      3,999.44        VA
                 09/02/98     1,653      6,775.55        OT

            Product: Stapler

Perrol, Barb     09/13/98     3,320     11,225.60        AC
Zonker, Pete     09/11/98       766        453.53        AC
                 09/20/98     1,455      2,543.43        VA
Perkins, Joe     09/12/98        23         23.55        OT
```

From this report you want to create a table that contains sales person, sales numbers, product and company, e.g. this table:

Company	Product	Sales Person	Date	Qty	Price	Class
XYZ Company	Tape	Doe, John	9/23/97	1120	11225.6	AC
XYZ Company	Tape	Doe, John	9/24/97	21123	233112.67	VA
XYZ Company	Tape	Doe, John	9/27/97	23	32.22	CS
XYZ Company	Tape	Wacks, Gene	9/1/97	3766	3455.55	AC
XYZ Company	Tape	Peters, Kate	9/30/97	4000	3999.44	VA
XYZ Company	Tape	Peters, Kate	9/2/97	1653	6775.55	OT
XYZ Company	Stapler	Perrol, Barb	9/13/97	3320	11225.6	AC
XYZ Company	Stapler	Zonker, Pete	9/11/97	766	453.53	AC
XYZ Company	Stapler	Zonker, Pete	9/20/97	1455	2543.43	VA

XYZ Company	Stapler	Perkins, Joe	9/12/97	23	23.55	OT

This import requires a model with three markers. The first marker is used to import the record information of Sales Person, Date, Qty, Price and Class fields. The second marker for the header Product, and the third to import the Company header. Follow these steps to create the model:

1.Select New from the File menu.

2.File to Import

In the file dialog select the report file “sample2.txt” from the Samples directory.

3.Select the type of file to import

Select Fixed, and click the Next> button.

4.Define Marker (first marker)

Edit the marker pattern and change it to:

" ##/##/##"

Make sure the “##/##/##” aligns with the dates in the file.

As you enter the marker, you will notice that lines the file are highlighted in yellow. These are the lines that match the marker pattern. Make sure that all the 10 detail lines are highlighted.

Note that the Marker Type is set to “Record”, this indicates that the matching this marker will create a new record (or row) in the import table each time it matches a line in the file.

5.Define Import Fields: (for the first marker)

Select text in the File window to define new fields for "Sales Person", "Date", "Qty", "Price", and "Class".

Make sure that the “Repeat” is set to Yes for the "Sales Person" field. This will repeat the sales person’s name for the records were the name is missing. E.g. "Doe, John", whose name occurs only in the first record, will be repeated in records 2 and 3 even though his name was not printed on the corresponding report lines.

For the “Qty” and “Price” fields select “Numeric” as field type.

6.Define New Marker Pattern: (second marker)

Click one of the lines with the product name in it, and press Yes to create a new marker. Alternatively, you can press the “New Marker” button to create a new marker.

Modify the marker pattern to:

" Product:"

Test that the pattern matches all the product lines by pressing the Refresh button.

For this marker pattern set the Record Type is to “Header”. This indicates that upon matching this marker Import Wizard will not create a new record (or row) in the import table. When a

match is found for a header marker only the fields associated with the marker are updated, in this case the “Product” field.

7.Define Import Fields: (for the second marker)

Select text in the File window to define the “Product” field. In the Field Properties window set “Repeat” to Yes to repeat the entry for every record.

8.Define New Marker Pattern: (third marker)

Create a new marker for the company field by clicking the line with the company name in it..

Modify the marker pattern to:

"Company:"

Test that the marker matches the company line and set the Marker Type to “Header”.

9.Define Import Fields: (for the third marker)

Select text in the File window to define "Company" field, again with “Repeat” set to Yes, then click the “Ok” button.

Other Import Examples

See also:
Import Example 1: Simple Import
Import Example 2: Header Imports
The Main Import Wizard Window

The examples are stored in the "Samples" subdirectory of the directory where you installed Import Wizard, "C:\Program Files\Import Wizard 8\Samples" by default.

Example 3: Address Import

This example shows how to solve problems with optional fields, i.e. fields that are not present in each record. By defining separate sections and markers for each field it is possible to correctly import such files.

Example 4: Delimited Text File

This example shows how to selectively import fields from a delimited text file. The file used in this example is the log file of a web server.

Example 5: Footer Import

Sometimes the required information is found in footers below the actual record. This example shows how to use footers in an inventory report.

Example 6: Use of Delimited Field No and Formula properties

This example uses the SAMPLE1.TXT file to create a table with the first/last name of the employee together with the total number of hours.

Delimited Field No/Delimiter properties

The employee name is in the source file in the format Last, First (for example: “Doe, John”) From this data the first and last name are extracted using the “Delim Field No” property. To extract the last name “Delim Field No” is set to 1 and “Delimiter” to {COMMA}, this will extract the first field from the left. For the first name “Delim Field No” is set to –1, thus extracting the first field from the right. Note that “Delim Field No” set to 2 would have worked as well of course.

Formula properties

The first and last name obtained using the method described above is combined into a field named “FirstLast Name” using the formula: First_Name & “ “ & Last_Name

Note the underscore (“_”) in the fieldnames in the formula. As spaces and other special characters are not allowed in formula names these have been replaced with an underscore. The name of a field for use in formulas is given by the Formula Field property.

The “Hours” field has this formula: Regular_Hours+Other_Hours+Overtime_Hours, thus adding up all the hours. Note that the “Regular Hours”, “Other Hours”, and “Overtime Hours” fields are present in the model but not output to the destination table. This is because the Skip property is checked for these fields.

Example 7: Multi Line Fields

This example demonstrates the use of the MultiLine Field Type property.

Example 8: Vertical Report

Sample report with records organized in columns instead of rows.

The Main Import Wizard Window

See also:
More Model Parameters Dialog

The main Import Wizard window shows the model properties of the current model.

All relative directory paths in this window are relative to the path of the model file. For example: the log file "log\mdl.log" for model "c:\models\mdl.iwm" will be created in the directory "c:\models\log\" and source file "..\src.txt" is located in "c:\".

Model File

Read only property, the file name of the current model definition. Selecting one of the filenames from the dropdown list will open that model file.

Model Description

You can use this property to attach notes to the model.

Source File(s)

The file(s) you want to import. You can use the “?” and “*” wildcards to select multiple files. It is also possible to supply multiple file names by entering each file name on a separate line. If no directory is specified, the previously selected directory will be used. Pressing the ... button will open the Browse file dialog, pressing the preview button will open the first source file in notepad.

The following example imports all files from the c:\dir1 directory and file1.txt and file2.txt from the c:\dir2 directory:

```
c:\dir1\*.*
c:\dir2\file1.txt
file2.txt
```

When a line starts with "-r " then files will be imported recursively from the specified directory and all its subdirectories. The following example imports html files from dir1 and all subdirectories of dir1:

```
-r c:\dir1\*.html
```

Connect String(Pro version only)

A string that defines the file type of the destination database is. Default is an empty Connect String, this will output the data to an Access database. Press the "Build" button build a connect string to for other database types or to output the results to a file. See Connect Strings for a description of the available output database and file types.

Database File(Pro version only)

The file name of the destination database. E.g. for Access databases enter the .mdb file, for Excel enter the .xls file, etc. Use the "..." button to browse for a file. Import Wizard can not create new database files, the database file has to exist before starting an import.

Table

The table in the destination database where you want to store the imported data. This can be an existing table or a new table. If you provide a new table name the Import Wizard will attempt to create the table. If the table exists then the existing table is used for the import and warnings are generated for field names that don't exist.

Note for Pro version only: Depending on the database type, creating a new table might not always be possible, in that case use the database program to define the table before importing.

Cell/Range(Excel version only)

The cell or range where the imported table has to be stored. You can use the navigator button (the button with a dash in it at the right edge of the Cell/Range input field) to select a cell in the spreadsheet. Press the button, select a cell, press the button again to copy the cell reference. By default this property is set to "A1" which will import into cell A1 of the current worksheet.

Filter

Leave blank to import all records, or enter a formula. The formula is evaluated just before a record is stored in the database. If the Filter formula evaluates to True (or a value not equal to 0) the record is stored in the database, otherwise the record is ignored. See Use of Formulas for details.

Import Action

Empty existing table: Delete all records from existing table before importing new records.
Append to table: Use existing table and append new records.
Create new table: Create a new table, even if a table already exists.
Insert: Adds records to end of the table.
Update: Replace records with matching Update Key fields.
Update, then Insert: First try to update, if no matching records is found then insert the record.

Note: when using the "Update" Import Action make sure that the update key fields are have an index defined in the destination database, otherwise the update operation will get very slow on larger tables. See benchmarks below, in this case an update with index is 15 to 50 times faster than an update without index. The Excel Add-In automatically generates an index

before importing.

Benchmark	Access	MySQL
Insert 10,000 records	1931 rec/sec	1066 rec/sec
Update 10,000 records without index	46 rec/sec	18 rec/sec
Update 10,000 records with index	670 rec/sec	878 rec/sec

Show Warnings

Check this box if you want to display import warning dialogs that might occur during import.

Log File

The filename where the import log is stored. The import log contains all import warnings, regardless of the “Show Warnings” setting. Leave the empty to skip log file creation. Pressing the ... button will open the Browse file dialog, pressing the preview button will open the log file in the text editor.

Select Import Type

Fixed select this option if the fields in the file appear on the same position on a line. See Define Fixed Import

Delimited select this option if the fields are on a single line separated by a specific character such as a comma or a tab. See Define Delimited Import

HTML Table select this option if the source file is a HTML file that contains tables. See Define HTML Import

Excel select this option if the source file is an Excel file. See Define Excel Import

Importing Fixed Position Files

See also:
Import Example 1: Simple Import
Import Example 2: Header Imports

Introduction

The screen is split in three windows:

File Window: The top half of the screen contains the File window which shows the file content. Lines in the File window that match the selected marker are highlighted in yellow. If a field is selected then the fields are also highlighted (in blue) in the File window.

Model Window: Bottom left is the Model window which gives an overview of the markers and fields defined in the model. When you select a marker or field in this window the File window highlights the part of the file that matches the selected item and the Properties window will show the properties for the selected item.

Properties Window: Bottom right is the Properties window which gives the property details for the selected marker or field in the Model window.

The File window supports two shortcuts. A new field is created when you select text in the File window (click and drag the mouse to select one or more characters). A new marker is created if you click a line in the File window.

The Copy, Delete, Up and Down buttons operate on the currently selected item in the Model window. For example: if the Copy button is pressed while a Field is selected then that field is copied into a new field.

Markers

You can create new markers by pressing the “New Marker” button Alternatively you can click a line in the File window to create a new marker with the line as marker pattern.

After you changing the marker pattern it is recommended to press the “Refresh” button to see if the marker actually performs as expected.

Markers have the following Marker Properties:

Marker Number

This number identifies the marker. It is recommended to make the marker with the lowest number the record marker. This because the import algorithm will try to match markers in

descending order, i.e. markers with the highest number first. Using a record marker with a higher number than a header marker might result in imports where header marker appears to be ignored.

It is allowed to define two or more markers with the same marker number. When doing an import the import fields for the marker number are updated when any of the markers with that marker number is matched. For example if you define two markers "#" and "A" both with marker number 1, then a match for marker number 1 is found when the first character of a line is either numeric (0-9) or "A".

Marker Pattern

The marker pattern can be edited directly at the top of the screen above the File contents display. Use the File vertical scroll bar to position the line you want to match directly under the pattern input box to help in aligning your pattern correctly.

The marker Pattern can contain multiple lines, a match is made when all the lines of the marker pattern match. The Pattern input box only displays a single line, use the up and down arrows to move to a different line. The current marker line position and the number of lines in the marker are displayed above the Pattern. Note: Discard marker patterns can only contain a single line, if a discard marker pattern contains more than one line only the first line of the pattern will be used.

The marker pattern can also contain regular expressions.

Examples:
Hello_@@ Matches a line that starts with "Hello", a space and two alpha characters.
first line: **<NEW LINE>** second line: **###** Matches a blank line followed by a line that start with three digits.

Marker Type

There are four marker types: Record, Header, Footer, and Discard.

Select **Record** if you want to create a new record when this marker is matched.

Select **Header** or **Footer** if you just want to update the associated fields.

Select **Discard** to filter out unwanted lines of the source file. Lines matching Discard markers are completely ignored in the actual import process, as if the line was not present in the source file. This marker type can be used to remove headers from a file where the actual records and headers are intermixed in such a way that the records can not be correctly matched. Any fields defined with a Discard marker will never be imported.

Line Offset

For discard markers only, defines the line offset of the line to discard to the matched marker line. For example a value of –2 will discard the line which is two lines above the matched

discard marker.

Fields

To create a field: select a part of a line in the File window. Alternatively, you can press the “New Field” button and change the Field Properties manually.

Press the “Refresh” button after changing a field property to update the File window for the new settings.

Field have the following Field Properties:

Name

This is the field name in the destination table. Make sure that the name follows the naming conventions of the database you are using. For example: field names in Excel should not contain a period “.”, an exclamation mark “!”, an accent grave “`” or square brackets “[]”.

Convert To, Conversion Method, and Multiline End

These properties specify the conversion rules to be used, see Convert To and Conversion Method Field Properties for details.

Repeat

When Repeat is switched on the program will "remember" the last imported value until a new value is imported. This function is useful for importing headers in order to repeat the header field information for each record under the heading. For Fixed imports Repeat comes in two types: "Repeat last imported value" and "Repeat when imported value is blank". **Repeat last imported value** repeats the last imported value even if this value was blank, this setting is useful for Header marker fields. **Repeat when imported value is blank** repeats the last value if the current value of the field is blank, this setting is useful for Record marker fields. See "Sample2 Fixed, Header markers" for an example of both types.

Start

The starting character position. The first position on a line is 1.

Length

The length in characters of the import field.

Line

The line offset of the import field relative to the line that matched the marker pattern. I.e. line 0 means the field is on the same line, -2 means the field is two lines above the matched line.

Marker Number

The marker number that will trigger the import of this field.

Delim Field No

The sub field number to import from the string extracted from the source file using the Length, Start, and Line properties. The delimited field number is counting from left to right for positive numbers, counting from right to left for negative numbers. I.e. –2 is the second sub field from the right, 1 is the left most field. For example the comma delimited string “aa,bb,cc” will return “cc” for delim field number 3 or –1.

Delimiter

The delimiter to use to subdivide the field. A delimiter can contain more than one character and any combination of characters and characters defined using the curly brackets. For example {SPACE}-{TAB} for fields delimited by a space,dash,tab.

Text Qualifier

If a field starts with this character then the next field is expected after a second occurrence of the Text Qualifier. Two consecutive Text Qualifier characters are treated as a single character upon import. The text qualifier is used to contain delimiter characters within a field. For example using a semicolon as delimiter for the string a;”b;c” will result in 3 subfields: a, “b and c” with an empty text qualifier, and 2 subfields: a and b;c with a double quote (“) as text qualifier.

Formula

If a formula is entered, the result of the formula will be stored in the field. For detailed information on formulas see Using Formulas

The formula can also contain a regular expression.

Formula Evaluation

This property controls the moment when the formula is evaluated. Possible settings are:

Before record append: The formula is evaluated before the field is stored in the table. Formulas are evaluated in a single pass in the order in which they appear in the Model window. For example if you have two fields fld_a and fld_b, with formulas **"X" & fld_b** and **"Y" & fld_a** respectively, importing data "A" in fld_a and "B" in fld_b will result in "XB" to be stored in fld_a and "YXB" in fld_b.

On marker match: The formula is evaluated at the moment when the marker for the field matches.

Formula Name (Read Only)

This read only property represents the field name for use in formulas. All non alpha-numeric characters in the field name are replaced by an underscore ("_"). Also, if the fieldname does not start with an alpha character an "a" is inserted in front of the fieldname.

Skip

If checked the field will not be stored in the database, useful for creating calculated fields that hold intermediate results.

Data Type

If Import Wizard creates a new table, the field will be created with this data type. Import Wizard does not change the structure of existing tables unless "Create Table" is specified as Import Action in the main window.

Update Key

When the "Update" Import Action specified in the main window this property determines which records are updated (replaced).

Importing Delimited Files

See also:
Delimiter Characters Definitions

Step 1. Select delimiter and text qualifier then press the Apply button

Delimiter

The delimiter character that separates fields in the import file. See Delimiter Characters Definitions

Text Qualifier

Field starting with this character are expected to end at the next occurrence of this character. I.e. embedded delimiter characters are ignored until the closing text qualifier character is processed. Two consecutive Text Qualifier characters are treated as a single character upon import. See Delimiter Characters Definitions

Apply button

Pressing this button will update the model with new fields for the selected setting of the Delimiter and Text Qualifier.

Step 2. Set field properties

Select a field from the "Fields" list, then set the field properties as required:

Name

The name of the selected field from the Field Names list.

Field No

The delimited field number to import from each line. This property in not used for fields with Convert To "Variable".

Convert To and Conversion Method

These properties determine the conversion rules to be used for the import of the field, see Conversion Field Properties for details.

Repeat

If set to "Yes" then the previous field value will be repeated if the current field value is empty.

Formula

If a formula is entered, the result of the formula will be stored in the field. For detailed information on formulas see Using Formulas

The formula can also contain a regular expression.

Formula Name (Read Only)

This read only property represents the field name for use in formulas. All non alpha-numeric characters in the field name are replaced by an underscore ("_"). Also, if the fieldname does not start with an alpha character an "a" is inserted in front of the fieldname.

Skip

If checked the field will not be stored in the database, useful for creating calculated fields that hold intermediate results.

Data Type

If Import Wizard creates a new table, the field will be created with this data type. Import Wizard does not change the structure of existing tables unless "Create Table" is specified as Import Action in the main window.

Update Key

When the "Update" Import Action specified in the main window this property determines which records are updated (replaced).

Importing HTML Files

With this option you can import one or more HTML tables. The screen is split up in four windows: top left is the “HTML File Structure” window, top right “Table Preview”, bottom left “Model” and bottom right “Field Properties”.

The “HTML File Structure” window shows an overview of the tables in the HTML file, clicking on a table will show that table in the “Table Preview” window. Pressing the “Apply” button will build a default model for this table, the result can be seen in the “Model” and “Field Properties” windows. Changing fields works in the same way as with a Defining a Fixed Import.

The “Table(s) to import” property can contain one or more tables. Tables are numbered sequentially as they occur in the HTML file. You can use commas to separate table numbers, and dashes to indicate ranges of tables. For example “1,4-6” will import tables 1, 4, 5, and 6.

The “Table Preview” window might contain the following entries ###Table:x###, ###Colspan###, ###Rowspan### to indicate that at that position a embedded table exists, or that the cell is covered in a column or rowspan. These entries will not show up in the actual imported table.

Field Properties

Select a field from the "Fields" list, then set the field properties as required:

Name

The name of the selected field from the Field Names list.

Field No

The delimited field number to import from each line. This property in not used for fields with Convert To "Variable".

Convert To and Conversion Method

These properties determine the conversion rules to be used for the import of the field, see Conversion Field Properties for details.

Repeat

If set to "Yes" then the previous field value will be repeated if the current field value is empty.

Formula

If a formula is entered, the result of the formula will be stored in the field. For detailed

information on formulas see Using Formulas

The formula can also contain a regular expression.

Formula Name (Read Only)

This read only property represents the field name for use in formulas. All non alpha-numeric characters in the field name are replaced by an underscore ("_"). Also, if the fieldname does not start with an alpha character an "a" is inserted in front of the fieldname.

Skip

If checked the field will not be stored in the database, useful for creating calculated fields that hold intermediate results.

Data Type

If Import Wizard creates a new table, the field will be created with this data type. Import Wizard does not change the structure of existing tables unless "Create Table" is specified as Import Action in the main window.

Update Key

When the "Update" Import Action specified in the main window this property determines which records are updated (replaced).

Importing Excel Files

Step 1. Select Sheet/Range then press the Apply button

Sheet/Range

The dropdown is filled with the Sheets (entries ending with $) and Named Ranges (entries without $) in the spreadsheet. Upon selecting an entry, the Table Preview will show the Excel table. You can also enter a value directly, for example Sheet1$b10:c20, to import only part of a sheet. If you enter a number here then the corresponding sheet will be imported. For example: '1' imports the first sheet from the Excel file.

Note: The the top-left cell and bottom-right cells of the table corresponds to the non-blank area of the selected sheet/range. For example, Sheet1 only contains data in cell B2 and C3, then Sheet1$ is a 2x2 table with top-left cell B2. This even holds true if an explicit range is specified: Sheet1$a1:d4 will still the same 2x2 table.

First row contains field names

When checked, the first line will be ignored when importing.

Apply button

Pressing this button will update the model with new fields for the selected setting of "Sheet/Range" and "First row contains field names".

Step 2. Set field properties

Select a field from the "Fields" list, then set the field properties as required:

Name

The name of the selected field from the Field Names list.

Field No

The field number (column) to import from each row. This property in not used for fields with Convert To "Variable".

Convert To and Conversion Method

These properties determine the conversion rules to be used for the import of the field, see Conversion Field Properties for details.

Repeat

If set to "Yes" then the previous field value will be repeated if the current field value is empty.

Formula

If a formula is entered, the result of the formula will be stored in the field. For detailed information on formulas see Using Formulas

The formula can also contain a regular expression.

Formula Name (Read Only)

This read only property represents the field name for use in formulas. All non alpha-numeric characters in the field name are replaced by an underscore ("_"). Also, if the fieldname does not start with an alpha character an "a" is inserted in front of the fieldname.

Skip

If checked the field will not be stored in the database, useful for creating calculated fields that hold intermediate results.

Data Type

If Import Wizard creates a new table, the field will be created with this data type. Import Wizard does not change the structure of existing tables unless "Create Table" is specified as Import Action in the main window.

Update Key

When the "Update" Import Action specified in the main window this property determines which records are updated (replaced).

What's New

Version 8.2.1

- Add: A number entered in the Excel import range name will import the corresponding sheet from the source file, e.g. "1" imports the first sheet in from the source Excel file.

Version 8.2.0

- Fix: Improved import speed for Add-in versions.
- Modify: COM Add-in renamed to IW8DLL, see VBA Functions.

Version 8.1.1

- Modify: Add additional code page selections to character set dropdown.
- Fix: Repeat field value in fixed models with more than one record marker did not work correctly.
- Fix: Closing a paused preview window kept process running in background.
- Fix: Tableviewer kept table open until application was closed, table is now closed when window is closed.

Version 8.1.0

- Add: Recursively import files from a directory and all its subdirectories, see Main Window.
- Add: Raw Code Page to import data byte based irrespective of system locale, see Config.
- Add: impwiz.CodePage function to convert raw byte data into specific code page, see Formulas.
- Modify: COM Add-in renamed from IW8DLL to ImpWiz81, see VBA Functions.

Version 8.0.2

- Add: Code Page translations, see Config
- Add: Automatic handling of Unicode source files.
- Fix: Command line parser did not work correctly.
- Fix: In fixed model design the repeat option labels were swapped.

Version 8.0.1

- First release of version 8

Version 8.0.0 (Beta Release)

- User interface improvements:
  - Add-In versions and Pro version now have uniform interface with toolbar buttons.
  - Source files: show *.* when selecting all files in a directory.
  - Show fields grouped by marker in fixed model design.
  - Last used model directory is default directory for file open/save.
- Update functionality to update (replace) existing records with new data based on matching key fields.

- Improved support for SQL-Server, MSDE, MySQL, Oracle, Interbase, Firebird and other databases.
- Connect String Builder to assist in setting up database connect strings.
- Passwords/Encryption
  - Open Password on model files prevents unauthorized access.
  - Modify Password on model files prevents unauthorized changes.
  - Password protected model files are encrypted to prevent unauthorized access via file editors.
- New field property "Data Type" specifies data type of field when table is created.
- New functions for use in formulas:
  - ProperCase: Convert string to proper case (i.e. with first letter of each word capitalized)
  - GetCityUS, GetStateUS, GetZipUS: Split US-Address into City, State and Zipcode parts
  - [fieldname].LastData retrieves field value in last record.

Version 7.0.11

Fix: Excel add-in did not import more than 32768 records.

Version 7.0.10

Fix: Problem with handling of empty fields in Excel Add-In (problem only existed in 7.0.9).

Version 7.0.9

Add: HTML import now replaces   entities with spaces.

Add: impwiz.IfNull function.

Fix: On some systems Excel add-in imported an empty table.

Version 7.0.8

Add: Add-ins for MS-Office 2003.

Add: “New line delimiter” in More Parameters to set non-standard line delimiters. See Config

Version 7.0.7

Fix: Add-In VBA function ImportWizardGo() showed “Do you want to save?” dialog when called with optional arguments.

Fix: Windows error on exit when Delim model designer was opened (happened only on Win2000 & ImpWiz Add-In).

Version 7.0.6

Fix: Filters on Fixed imports with fields that do not occur in every record.

Add: New field property “Data Type” to define datatype of created database tables for Delim imports.

Version 7.0.5

Fix: Next button on first New Model dialog does not always display second dialog.

Add: Multiline field support for delimited imports.

Add: Environment variables formatted as %USERNAME% allowed in Source, Log, and DB filenames.

Version 7.0.4

Add: Instructions to create user-interface translations (See International Versions)
Fix: HTML import problem
Fix: Import into SQL-Server 2000 problem.
Add: Spanish and Italian interface translations.

Version 7.0.3

Add: New field property “Formula Evaluation” to control when a formula is evaluated. See New/Modify Model - Define Fixed Import
Add: New marker string <LINENO n,m> to match specific line numbers. See Marker Pattern Definitions
Fix: In fixed import creating multiple records from a single line works again. (Was working in version 6.0.9)
Add: Option to specify text editor command line.
Fix: Excel Add-in keeps text format. For example text import of “01MAR” is displayed as is, it used to display as “1-Mar”

Version 7.0.2

Add: Support for source files in DOS character set.
Add: Fixed line lengths.
Add: First byte offset.
Fix: Access Add-Ins progress display.
Fix: Excel 97 Add-In startup problem.

Version 7.0.1

First release of version 7.

Version 7.0.0 (Beta Release)

Add: Convert To and Conversion Method field properties are preset to Text, Date or Numeric.
Add: Option to select font for Model Design dialog.
Modify: Fixed Model Design dialog displays full 32000 characters per line, and fixed display font can be chosen to display a larger portion of the file.
Modify: Returnvalues of VBA functions and commandline call simplified.
Add: Export (output) imported table to XML files.
Add: New variables RecordID, TotalRecordID, and TotalRecordNo.
Fix: Delimited imports can now have multiple fields with the same delimited field number.
Add: Holding Field/Marker Up or Down arrow will continuously move the Field/Marker
Add: In Fixed model design, markers updates refresh display automatically
Add: “Preview Import” function to preview import results without actually importing.
Modify: Model files are now saved in XML format. Version 7 opens previous model file formats, but model files will be saved in the new format.
Modify: update to msvbvm60.dll from msvbvm50.dll

Version 6.0.9

Fix: Display control characters as ~ in model design window.

Version 6.0.8

Modify: Fixed Model Design dialog displays lines up to 2000 characters (was 500), see Import Algorithm and Limitations.
Fix: System lockup in Footer markers. This bug appeared in 6.0.7, earlier versions were fine.

Version 6.0.7

Add: Most recently used files in can be accessed via model file dropdown.
Modify: Delimited fields, fields starting with a text qualifier are now truncated at the first non-escaped text qualifier. (Before the field was truncated at the next delimiter.)
Modify: Increase line offset and number of multilines from 100 to 255.
Fix: Some Fixed Model Design features did not work properly.
[Pro only] Add: when used in batch mode impwiz.exe returns the result of the import.

Version 6.0.6

Fix: Upon creating a field in fixed mode, the file display jumps to first character of the line instead of showing the created field.
Fix: Upon modifying a model after deleting fields, the Model subwindow was not always displayed correctly.
[Access Only] Fix: sourcefile and logfile were not copied into model after using file-browse button.

Version 6.0.5

Add: HTML imports <TH> tags as well.
Add: 'Repeat within marker' field property.
Fix: Increase max chars in source file(s) dialog from 256 to 2048.

Version 6.0.4

Add: HTML table import, see New/Modify Model - Define HTML Import.
[Pro only] Add: HTML, Excel and dBase file output, see Output to Files (Pro Version Only).
Add: Formula and Skip field properties to delimited import.
[Pro only] Add: MS SQL Server support, see Database Types (Pro Version Only).

Version 6.0.3

Add: Multiline markers
Add: Trim/Raw and NewLine keywords for MultiLine formats.
Fix: Import warnings for incorrect dates (was working in version 5.1).

Version 6.0.2

Fix: Table field type was always set to text on new tables.
Fix: Filter condition is now evaluated after evaluation of variables, regular expressions, and

formulas.
Fix: Repeat field property didn’t work correctly on delimited imports.
Modify: Install/uninstall program by www.jrsoftware.org.

Version 6.0.1

First release of version 6, includes some minor bug fixes.

Version 6.0.0 (Beta Release)

Improved model design interface: highlights fields, markers and discard markers in the source file.
Modify field start, length, and line properties by CTRL-selecting text in the source file.
Line offset property for discard markers.
The same fields can be defined more than once which allows for imports with more than one record of a single line.
Regular expressions in markers and formulas.
The same field can be defined more than once to enable imports with multiple records on a single line.
Multi lingual interface: English, French, and German.
Optimized code for improved speed.
Markers are processed in the order that they appear in the design window (was descending order of marker number).
Single setup program for executable and add-in versions.
Version 6.0 models are upwards compatible with version 5.x models.

Import Algorithm and Limitations

Import Algorithm

Import Wizard will use the model definition in the following way to parse the source file into a table.

Step 1: Read next line from the source file. If the line matches a Discard marker then repeat Step 1, other wise continue at Step 2.

Step 2: Match line with markers. The line from Step 1 is matched against the first marker. If the marker matches the line then the fields defined for that marker are read into a temporary import record. If the marker does not match then the next marker is processed. Upon finding a match for a Record marker the following three steps occur:

i. The fields for the record marker are read into the temporary import record.
ii. The formulas are evaluated. Evaluation occurs in a single pass so that circular references are not possible.
iii. The temporary import record is appended to the import table.

When all markers are processed the import continues with Step 1, the import is finished when all lines from the import file have been processed.

Note: In version 5 and before the markers were processed in descending Marker Number order, from version 6 onward the markers are processed in the order that they appear in the model.

Limitations

Number of records: Not limited by Import Wizard, limited by the database system used. For example, imports into Excel are limited by the maximum number of rows allowed in a spreadsheet.

Source file size: Unlimited for fixed and delimited imports (the file is read part by part). Limited by system memory size for HTML imports (the whole file is read into memory and then processed).

Source file line length: Up to 32767 characters for fixed and delimited imports. Unlimited for HTML imports.

Number of fields: Unlimited for fixed imports, 1000 for delimited imports.

Number of markers: Unlimited.

Multiline fields: Up to 255 lines.

Field Line offset: Line offsets allowed between -255 and +255 lines.

Model Design: The first 2000 lines of the file are displayed.

Discard Marker Pattern: Limited to a single line. If the discard marker contains more than one line then only the first line is used.

Discard Marker Line offset: Line offsets allowed between -255 and +255 lines.

Output to files using "File;" connect strings: See Connect Strings (Pro Version Only)

Marker Pattern Definitions

The following characters and strings have special meaning when used in a marker pattern:

Character	Matches which characters in the source file
Space	Any character
?	Non space characters
@	Alpha (a-z and A-Z) characters
#	Digit (0-9) characters
_	Space character

String	Matches what source file lines
<New Line>	An empty line or a line containing only spaces
<New Page>	A line containing the form feed character (ASCII code 12)
<LineNo n,m>	Matches line number n and repeat every m lines. Parameter m is optional. Example: <LineNo 1,4> will match the 1st, 5th, 9th, 13th, etc. lines of the source file.

Note: If the marker pattern starts with a slash (“/”) then the pattern is interpreted as a Regular Expression.

Regular Expressions

Regular Expressions can be used in marker string and formula field properties. To enter a regular expression enter a forward slash, followed by the options, followed by another forward slash and the regular expression itself:

/<options>/<regular expression pattern>

The two forward slashes are mandatory, even if no options are given. The first forward slash has to be at the first character position, otherwise the string will not be interpreted as a regular expression.

The **<options>** are:

i Matching is case sensitive by default, including the "i" option makes the regular expression match is case insensitive. Note that character ranges are always case sensitive, regardless of this setting.

0-9 When used as field formula property, will return substring number. 1 will return the first substring, 2 the second, etc. 0 will return the full match of the regular expression. (Default is 0)

The **<regular expression pattern>** is the pattern to match. In the pattern you can use the following:

^ Matches the beginning of input or line.

$ Matches the end of input or line.

Matches the preceding character zero or more times. "zo" matches either "z" or "zoo." **+** Matches the preceding character one or more times. "zo+" matches "zoo" but not "z."

? Matches the preceding character zero or one time. "a?ve?" matches the "ve" in "never."

. Matches any single character.

(pattern) Matches pattern and remembers the match. The matched pattern can be retrieved using 0-9 in <options>.

x|y Matches either x or y. "(z|f)ood?" matches "zoo" or "food".

[xyz] A character range. Matches any one of the enclosed characters. "[abc]" matches the "a" in "plain." Use of ranges is allowed: "[a-zA-Z0-9]" matches alpha-numeric characters.

[^xyz] A negative character range. Matches any character not enclosed. "[^abc]" matches the "p" in "plain."

\ Escape character, used to indicate that the next character is not a pattern but a normal character. This also means that you have to write "\\" to represent a backslash. The escape character is not used in a character set. "a*" matches "a*" but not "aa".

other Any other single character with no other significance, matches that character.

Regular Expression Syntax

A regular expression is one or more branches, separated by "|". It matches anything that matches one of the branches.

A branch is zero or more pieces, concatenated. It matches a match for the first, followed by a match for the second, etc.

A piece is an atom possibly followed by "*", "+", or "?". An atom followed by "*" matches a sequence of 0 or more matches of the atom. An atom followed by "+" matches a sequence of 1 or more matches of the atom. An atom followed by "?" matches a match of the atom, or the null string.

An atom is a regular expression in parentheses (matching a match for the regular expression), a range (see below), "." (matching any single character), "^" (matching the null string at the beginning of the input string), "$" (matching the null string at the end of the input string), a "\" followed by a single character (matching that character), or a single character with no other significance (matching that character).

A range is a sequence of characters enclosed in "[]". It normally matches any single character from the sequence. If the sequence begins with "^", it matches any single character not from the rest of the sequence. If two characters in the sequence are separated by "-", this is shorthand for the full list of ASCII characters between them (e.g. "[0-9]" matches any decimal digit). To include a literal "]" in the sequence, make it the first character (following a possible "^"). To include a literal "-", make it the first or last character.

Ambiguity

If a regular expression could match two different parts of the input string, it will match the one which begins earliest. If both begin in the same place but match different lengths, or match the same length in different ways, life gets messier, as follows.

In general, the possibilities in a list of branches are considered in left-to-right order, the possibilities for "*", "+", and "?" are considered longest-first, nested constructs are considered from the outermost in, and concatenated constructs are considered leftmost-first. The match that will be chosen is the one that uses the earliest possibility in the first choice that has to be made. If there is more than one choice, the next will be made in the same manner (earliest possibility) subject to the decision on the first choice. And so forth.

For example, "(ab|a)b*c" could match "abc" in one of two ways. The first choice is between "ab" and "a"; since "ab" is earlier, and does lead to a successful overall match, it is chosen. Since the "b" is already spoken for, the "b*" must match its last possibility--the empty string--

since it must respect the earlier choice.

In the particular case where the regular expression does not use "|" and does not apply "*", "+", or "?" to parenthesized subexpressions, the net effect is that the longest possible match will be chosen. So "ab*", presented with "xabbbby", will match "abbbb". Note that if "ab*" is tried against "xabyabbbz", it will match "ab" just after "x", due to the begins-earliest rule. (In effect, the decision on where to start the match is the first choice to be made, hence subsequent choices must respect it even if this leads them to less-preferred alternatives.)

Examples

Marker String: "//[0-9]+/[0-9]+/[0-9]+"

Matches: "12/31/2001" or "The last date was 2/7/99" but not "1-31-01"

Field Formula: "/i/([0-9]+):([0-9]+) *[ap]m"

Returns: "9:45 PM" from “The time is 9:45 PM”

Field Formula: "/i1/([0-9]+):([0-9]+)"

Returns: "9" from “The time is 9:45 PM”

Field Formula: "/i2/([0-9]+):([0-9]+)"

Returns: "45" from “The time is 9:45 PM”

Delimiter Character Definitions

The following formats can be used to define a Delimiter character, a Text Qualifier string, and the LineBreak multiline Format property:

Format	Description
{TAB}	Tab (ASCII code 9)
{SPACE}	Space
{NULL}	Null character (ASCII code 0)
{COMMA}	Comma (,)
{NONE}	Nothing (not valid as delimiter character)
{DBLQUOTE}	Double quote (“)
{SGLQUOTE}	Single quote (“)
{COLON}	Colon (:)
{SEMICOLON}	Semicolon (;)
{x}	With x a decimal number between 0 and 255, defines a character with ASCII code x

Formulas

Formulas are used in the Formula field property and in the Filter model property.

You can use any VBScript expression in your formula, such as Mid(), Left(), DateSerial(), see below for a summary of VBScript operators and functions. Use the name specified in the read-only field property Formula Name to refer to the content of a field. The Formula Name property translates the Name of the field to a name without spaces or special characters so that it can be used in formulas. The value of the field in the previous record can be accessed with the LastData property. Example for a field with name "Field1": the formula Field1 return the value of the field, and the formula Field1.LastData will return the value of the field in the previous record.

Examples:

Filter

Left(Some_Field,3)<>”Doe”

This will import only records where the first 3 characters of the field “Some Field” are not equal to “Doe”

Field Formula

“S” & Right(“000000” & Sales_Order_No,6)

This will convert field “Sales Order No” to a string starting with an “S” followed by 6 digits, i.e. 56 is converted to “S00056”.

DateSerial(1999,12,31)

VBScript Operators and Functions Summary

This summary contains the for data conversion most useful VBScript operators and functions. Arguments between square brackets [] are optional.

Operator Summary

&

Used to force string concatenation of two expressions.

^ * / Mod + -

Mathematical operators: raise a number to the power of an exponent (^), multiply two

numbers (*), divide two numbers (/), divide two numbers and return only the remainder (Mod), sum two numbers (+), find the difference between two numbers or to indicate the negative value of a numeric expression (-).

< <= > >= = <>

Comparison operators: less than (<), less than or equal to (<=), greater than (>), greater than or equal to (>=), equal to (=), not equal to (<>).

And Or Not

Logical operators: logical conjunction on two expressions (And), logical disjunction on two expressions (Or), logical negation on an expression (Not).

String Functions Summary

Lcase(string), Ucase(string)

Convert string to lowercase or uppercase.

Len(string)

Find length of a string.

Mid(string, start[, length]), Left(string, length), Right(string, length)

Return a substring of string.

InStr([start,] strHaystack, strNeedle)

Returns the position of the first occurrence of strNeedle in strHaystack. The optional argument start sets the starting position for each search. If omitted, search begins at the first character position. Returns 0 if no match was found.

LTrim(string), RTrim(string), Trim(string)

Returns a String containing a copy of a specified string without leading spaces (LTrim), trailing spaces (RTrim), or both leading and trailing spaces (Trim).

Asc(string)

Returns an Integer representing the character code corresponding to the first letter in a string.

Chr(charcode)

Returns a String containing the character associated with the specified character code.

Date/Time Functions Summary

DateSerial(year, month, day)

Returns a Date for a specified year, month, and day.

DateValue(date)

Returns a Date. The required date argument is normally a string expression representing a date from January 1, 100 through December 31, 9999. However, date can also be any expression that can represent a date, a time, or both a date and time, in that range.

Year(date), Month(date), Day(date), Hour(date), Minute(date), Second(date)

Returns an Integer specifying respectively the year, month, day, hour, minute, or second as of the specified date.

Now, Date, Time

Returns current date and time (Now), date (Date) or time (Time) according your computer's system date and time.

Math Functions Summary

Int(number)

Returns a value of the type passed to it containing the integer portion of a number.

Abs(number), Atn(number), Cos(number), Exp(number), Log(number), Sin(number), Sqr(number), Tan(number)

Return result of the math function applied.

Rnd[(seed)]

The Rnd function returns a random number less than 1 but greater than or equal to zero.

Import Wizard (impwiz) Extension Functions

impwiz.IIf(expression, truepart, falsepart)

Inline if function, if expression evaluates to true then truepart is returned, otherwise falsepart is returned. Note that this function fails if either expression, truepart, or falsepart contains an error.

Example: impwiz.IIf(overtime_hours>0,"OVERTIME","NO OVERTIME")

impwiz.IfNull(expression, nullvalue)

Returns nullvalue if expression is Null, otherwise this function returns expression.

Example: impwiz.IfNull(overtime_hours,0)
This will replace empty (null) overtime values with a 0 (zero).

impwiz.GetUSCity(address)

Returns the city name from a US address line formatted as "city, state zipcode", or an empty string if the address could not be parsed.

Example: impwiz.GetUSCity("Glendale, AZ 85306")
Gives "Glendale" as result.

impwiz.GetUSState(address)

Returns the state name from a US address line formatted as "city, state zipcode", or an empty string if the address could not be parsed.

Example: impwiz.GetUSState("Glendale, AZ 85306")
Gives "AZ" as result.

impwiz.GetUSZip(address)

Returns the zip code from a US address line formatted as "city, state zipcode", or an empty string if the address could not be parsed.

Example: impwiz.GetUSZip("Glendale, AZ 85306")
Gives "85306" as result.

impwiz.CodePage(bytestring, codepage)

Converts the binary string bytestring, for example a field imported with codepage raw, to the specified codepage.

Import Error and Warning Messages

During imports error or warning messages might pop up. Error messages are identified by an error code starting with an **E**, these errors will terminate the import. Warning messages are identified by an error code starting with **W**, these warnings will not terminate the import, but the import might not be complete. Most warning messages relate to fields that can not be translated into a specific field type. For example if you defined a field as numeric you will get a import warning message if you try to import non numeric characters into this field. By changing the field type to "Text" which will take any data, you will not see this particular warning again.

The following errors or warnings might occur:

W001: Source file not found: <Source File Name>
Upon trying to import using the model Import Wizard couldn't find the source file. Supply the correct file name.

E002: Source file name not supplied.
The source file was not supplied in the model. Supply a source file name.

W004: Error in date conversion of "<Import Data>" with parse string "<Format String>"
Import Wizard is unable to convert <Import Data> to a date using the format supplied in the model. Correct the format in the model and try again.

W005: "<Error Description>" in field "<Import Field Name>", while importing: "<Import Data>"
A general error occurred while importing <Import Data> to <Import Field Name>. Modify your model to correct this.

W006: Error "<Error Description>" while appending record.
A general error occurred while attempting to append a record to the destination table, the record is not appended.

W007: Import cancelled by user.
The import was aborted by pressing the cancel button in the progress dialog.

W008: This software is unregistered, ONLY PART OF THE FILE(S) WILL BE IMPORTED.
Please enter your registration code via Enter Registration Code option in the Help menu. See Ordering Software, Registration to enable unlimited imports. (The import progress will continue without actually storing records to demonstrate software capabilities.)

E011: Model "<Model Name>" not found.
An import was attempted with an unknown model name.

E012: Database connection failed: <Error Description>

Import Wizard was unable to connect to the destination database.
Actions:
1) Check that all required components are installed, see Installation.
2) Check that the entries in the main screen are correct.
3) Check that the database is not exclusively locked by another user, or read only.
4) When using a ODBC database check that the connect string is correct, common mistakes include incorrect passwords, or user accounts with insufficient rights. See Database Types (Pro Version Only).

W013: Field "<Field Name>" not found in destination table.
The destination table exists but does not contain the <Field Name> field. Imports to this field will be ignored.
Action: either modify the destination table in your database to include the field or import to a new table and let Import Wizard build the table definition in your destination database.

E014: Destination table connection failed: <Error Description>
The destination database could be opened but the destination table couldn't.
Action: When using Excel, make sure that a correct range name or cell reference is entered in the "Table" property, in doubt use "A1" as the table name as this should always work. Also make sure you have sufficient rights to the destination table.

W015: Failed to delete current records in table: <Error Description>
The destination table contains records and you specified no Append, but the destination database does not support the delete action, an example of such a database is an Excel spreadsheet.
Action: check the Append check box, and delete the table contents in the destination database before importing if required.

E016: Table "<tablename>" not updateable
The table could not be created because it is marked as read-only.

W017: Unable to create field "<fieldname>", please check model.
The field could not be created. Check for illegal characters in the fieldname.

E018: Unable to create table "<tablename>", please check model or create table manually.
There was a general error trying to create the table.

E019: Destination output file "<filename>"could not be created: <Error Description>
The output file defined in the model could not be created. Check that the name is valid and that you have sufficient rights to create the file.

W020: Marker number does not exist. Defined in multline field: "<FieldName>", Marker no: "<MarkerNo>", Multi Line End: "<MultiLineEnd>"
The marker referenced in the multiline end property of field fieldname does not exist. Edit this

property to correct.

W021: Invalid marker number. Defined in multline field: “<FieldName>“, Marker no: “<MarkerNo>“, Multi Line End: “<MultiLineEnd>“
The marker referenced in the multiline end property of field fieldname is invalid. Valid marker numbers are integers larger than 0. Edit this property to correct.

W022: Error in filter: <ErrorDescription>, the error occurred in column <Column>
The Filter contains an error at character position Column. The Filter will be ignored and all records will be imported. Edit the Filter to fix this problem.

W023: Error in formula of field “<FieldName>“: <ErrorDescription>, the error occurred in <Column> of formula “<Formula>“
The formula defined for this field contains an error. The formula will be ignored.

E024: Script error
A non-trappable script error occurred. Unfortunately it is not possible to trace this error back. Check the formulas of fields and the filter formula for errors. If this doesn’t help, start removing formulas until the error does not occur anymore. The last removed formula caused the error.

E025: Script timeout
A script timeout occurred, possible due to an endless loop. Check the formulas of fields and the where condition. If this doesn’t help, start removing formulas until the error does not occur anymore. The last removed formula caused the error.

W026: Multiline field “<FieldName>“ truncated to <n> lines.
The multiline end condition for the field was not met before reaching n lines. The only the first n lines of the multiline field will be imported. Check the multiline end condition, see also Limitations.

W027: Undefined marker number “<MarkerNo>“ in field “<FieldName>“.
The field contains a reference to an undefined marker number. Either create a new marker for the missing marker number or change the Marker No property of the field.

W028: VBScript object initialization failed, formulas will not be available.
Check that the file msscript.ocx is present in the system directory (c:\windows\system\ or c:\winnt\system32\ by default). If it isn’t present, copy the file from the setup distribution. Register the file by opening a command prompt, changing to the system directory and typing the command: “regsvr32 msscript.ocx”, you should get a message that the file was registered successfully.

W029: RegExp object initialization failed, regular expressions will not be available.
Check that the file sgregexp.dll is present in the install directory (c:\Program Files\Import Wizard 6.0\ by default). If it isn’t present, copy the file from the setup distribution. Register the file by opening a command prompt, changing to the install directory and typing the command: “regsvr32 sgregexp.dll”, you should get a message that the file was registered

successfully.

E030: Output table does not support Append. Clear Append in the model and try again.

It is not possible to append records to the output table. To import the file(s) uncheck the “Append” checkbox of the modelproperties and try again.

W031: This model file was created with a newer version of Import Wizard.

The model will still load, but any model features of the newer version will be ignored when importing or saving the model. If you do not want the lose the model features of the newer version, download and install the latest version of Import Wizard.

W032: Error <Error Description> while opening file '<FileName>

The file could not be opened. It might be locked exclusively by another program, or you might not have sufficient priveleges to open the file.

W033: Import Action Warning: <Warning Description>

There was a problem with the Import Action specified, the Warning Description gives details and solution.

E034: Import Action Error: <Error Description>

There was a major problem with the Import Action specified, the Error Description gives details and solution.

E035: Destination cell or range not found <Rangename>

The spreadsheet destination cell/range was not specified or incorrect. Correct the entry and try again.

W036: Code Page <CodePage> not found. Code Page Config setting will be ignored.

The Code Page specified in Config is not valid, change the setting and try again.

W037: Code Page <CodePage> conversion error. Code Page Config setting ignored.

Converting (a part) of the source file with the indicated Code Page failed.

VBA Functions (Access & Excel Add-in Versions Only)

Import Wizard exposes two VBA functions that can be used to automate your application. The functions are: ImportWizardGo() to start an import, and ImportWizardStart() to show the main Import Wizard window. The function are accessed as methods of the Application.COMAddIns("IW8DLL.ImportWizardAddIn").Object object. It is probably the easiest to define a new object variable and use this variable to access the functions:

```
Dim iw As Object
Set iw = Application.COMAddIns("IW8DLL.ImportWizardAddIn").Object
iw.ImportWizardStart
```

Function ImportWizardGo(ModelFile, Optional SourceFile, Optional DestTable, Optional Action) as Integer

This function will perform an import using the model defined in ModelFile. Calling the function with a value for any or all of the three optional arguments (SourceFile, DestTable, Append) will override the setting in the model.

Action sets the import action(s) to perform.
1: Create new table, remove existing table and recreate it
2: Empty table, clear all existing records from the table before importing.
4: Insert, add imported records to the end of the table.
8: Update, replace existing records with imported records when the update key fields match
To add the numbers to combine actions, for example the value 5 (=1+4) will create a new table and then insert the records. If neither 1 or 2 is specified the records will be appended to an existing table.

Return values:
0: Import completed without warnings or errors
1: Import completed with warnings.
2: Import failed, an fatal error occurred.

Example: import the sales2005.txt file using the Sales.iwm model and display a message if import had warnings and/or errors.

```
returnvalue = ImportWizardGo("Sales.iwm", "sales2005.txt")
If returnvalue > 0 then Msgbox "Import problems…"
```

ImportWizardStart(Optional ModelFile, Optional SourceFile, Optional DestTable, Optional Action)

This function opens the main Import Wizard window. If the ModelFile argument is set, the model will be opened but the import will not be started. Calling the function with a value for any or all of the three remaining optional arguments (SourceFile, DestTable, Append) will override the setting in the model.

Database Connect Strings (Pro Version Only)

The Connect String defines the type of output that is generated by Import Wizard. Output can be directed to a database, a spreadsheet, or a file. The file based "File;Type=..." Connect Strings for Delimited, dBase, Excel, HTML, SQL Script, and XML output are fully implemented within Import Wizard and do not require any external driver to be installed. For the other Connect Strings you might have to obtain drivers from your database supplier before you can use Import Wizard.

The following database, spreadsheet, and file types are supported:

Access
dBase File
Delimited File
Excel File
HTML File
MySQL
MSDE - SQL Server Desktop Edition
SQL Script File
SQL Server
XML File
Other Databases (ODBC)

Output to Files

Connect strings starting with "File;" enable output various types of files without using any external drivers. Supported are the following files types: delimited text, HTML, XML, Excel, SQL script, dBase. The filename of the output file is given in the Database File field. The Database Table field is not used.

The "File;" connect string can be followed by tags to further specifying the output file type. A tag has the format "TagName=tagvalue;", the tag starts with the tag name followed by an equal sign (=), followed by the tag value and is terminated with a semicolon (;). All tags are optional and can be specified in any order. If a tag is not specified in the connect string, the default value for that tag will be used.

The type of the output file is selected with the Type tag. If the Type tag is missing or incorrect the output format will default to Delimited Text.

Connect String Examples

File;Type=Delimited;NumericFormat=0.00;FieldNames=True

Comma delimited file numeric values formatted to two digits behind the decimal point, the first row will contain the field names.

File;Type=Delimited;Delimiter={tab};TextQual={none};DateFormat=yymmdd

Tab delimited file with no text qualifier and date formatted as two-digit year, two-digit month, and two-digit day.

File;Type=HTML;FieldNames=True;NumericFormat=0.0000

Creates a HTML file that contains table. The first row of the table contains the field names. All numeric values in the table are output with 4 digits behind the fractional mark.

File;Type=Excel;FieldNames=True;

Creates an Excel spreadsheet file with the imported table. The first row of the table contains the field names.

Access

This is default database for Import Wizard. Leave the connect string blank to import into any Access database version. By specifying an explicit Access version in the connect string, for example “Access 2000;”, Import Wizard will create the specified Access database version if the database file does not exist.

Note: To import into Access databases we recommend using the appropriate Import Wizard add-in version for Access. The Add-In versions for Access, “Import Wizard Access 97” and “Import Wizard Access 2000” do not depend on external database drivers and can be invoked from the Access main menu. You can also automate your imports with the Import Wizard VBA, see Visual Basic Functions (Add-in Versions Only).

dBase File

The format of the connect string for dBase files is:

File; Type=dBase;

The resulting dBase (.dbf) file can be read by dBase version III and higher.

Limitations: Text fields are limited to 254 characters. Field names are limited to 11 characters, longer names will be truncated.

Delimited File

The format of the connect string for delimited text files is:

File; Type=Delimited;
Delimiter=<delimiter>;
TextQual=<textqual>;
FieldNames=<True|False>;
DateFormat=<dateformat>; NumericFormat=<numericformat>;

Tags:

Type=Delimited;
Specifies delimited text output.

Delimiter=<delimiter>;
This is a single character delimiter. Default value is {comma}, see Delimiter Characters Definitions

TextQual=<textqual>;
This is a single character text qualifier. Text fields will be embedded between this character. Embedded text qualifiers in an output string will be replaced by two text qualifier characters. Default value is {dblquote}, see Delimiter Characters Definitions

FieldNames=<True|False>;
If set to True the first row in the output file contains fieldnames and is ignored when importing. Default value is False

DateFormat=<dateformat>;
Set formatting for dates. See below for detailed specification.

NumericFormat=<numericformat>;
Set formatting for numbers. See below for detailed specification.

Excel File

The format of the connect string for Excel files is:

File; Type=Excel;
FieldNames=<True|False>;

The resulting Excel (.xls) file can be read by Excel version 2.1 and higher.

Limitations: Text fields are limited to 255 characters and the file can contain 65535 records maximum.

Tags:

Type=Excel;
Specifies Excel output.

FieldNames=<True|False>;
If set to True the first row in the output file contains fieldnames and is ignored when

importing. Default value is False

HTML File

The format of the connect string for HTML files is:

File; Type=HTML;
TextQual=<textqual>;
FieldNames=<True|False>;
DateFormat=<dateformat>;
NumericFormat=<numericformat>;

Tags:

Type=HTML;
Specifies HTML output.

TextQual=<textqual>;
This is a single character text qualifier. Text fields will be embedded between this character. Embedded text qualifiers in an output string will be replaced by two text qualifier characters. Default value is {dblquote}, see Delimiter Characters Definitions

FieldNames=<True|False>;
If set to True the first row in the output file contains fieldnames and is ignored when importing. Default value is False

DateFormat=<dateformat>;
Set formatting for dates. See below for detailed specification.

NumericFormat=<numericformat>;
Set formatting for numbers. See below for detailed specification.

MySQL

MySQL is supported via the MyODBC driver available from www.mysql.com.

Field names with spaces are not allowed. The table and fields will be created, but an import will fail.

Some MyODBC version do not work with Import Wizard, in particular version 3.51.10-2 is known not to work. If you encounter problems with importing you could try uninstalling your MyODBC driver and install version 3.51.10 which has been tested to work correctly. You can download version 3.51.10 from: http://www.beside.com/download/MyODBC-3.51.10-win.msi

MySQL connect string:
ODBC;DRIVER={MySQL ODBC 3.51 Driver}; Selects MySQL driver
Uid=username; Username

Pwd=password; Password

DB=database; Database Name, mandatory, do not leave blank!

Server=localhost; Server hostname (default=localhost)

Port=3306; Server port (default=3306)

Option=3; Options, 3 is required for inserts to work!

MySQL SQL settings:

Name Escape: ` (backtick)

TextEscape: / (backslash)

DateFormat: \'yyyy-mm-dd hh:nn:ss\'

Created field types:

```
Text - VARCHAR(255)
Memo - MEDIUMTEXT
DateTime - DATETIME
Double - DOUBLE
Long - INT(11)
```

The following table list the option values available. Sum the values to set multiple options, and always set option 1+2. For example to set compressed transfers use "OPTION=2051" = 2048 + 2 + 1.

Value	Description
1	The client can't handle that MyODBC returns the real width of a column.
2	The client can't handle that MySQL returns the true value of affected rows. If this flag is set, MySQL returns ``found rows'' instead. You must have MySQL 3.21.14 or newer to get this to work.
4	Make a debug log in `c:\myodbc.log'. This is the same as putting MYSQL_DEBUG=d:t:O,c::\myodbc.log in `AUTOEXEC.BAT'. (On Unix, the file is `/tmp/myodbc.log'.)
8	Don't set any packet limit for results and parameters.
16	Don't prompt for questions even if driver would like to prompt.
32	Enable or disable the dynamic cursor support. (Not allowed in MyODBC 2.50.)
64	Ignore use of database name in db_name.tbl_name.col_name.
128	Force use of ODBC manager cursors (experimental).
256	Disable the use of extended fetch (experimental).
512	Pad CHAR columns to full column length.
1024	SQLDescribeCol() will return fully qualified column names.
2048	Use the compressed client/server protocol.
4096	Tell server to ignore space after function name and before `(' (needed by PowerBuilder). This will make all function names keywords.
8192	Connect with named pipes to a mysqld server running on NT.
16384	Change LONGLONG columns to INT columns (some applications can't handle LONGLONG).
32768	Return 'user' as Table_qualifier and Table_owner from SQLTables (experimental).

65536	Read parameters from the [client] and [odbc] groups from `my.cnf'.
131072	Add some extra safety checks (should not be needed but...).
262144	Disable transactions.
524288	Enable query logging to `c:\myodbc.sql'(`/tmp/myodbc.sql') file. (Enabled only in debug mode.)
1048576	Do not cache the results locally in the driver, instead read from server (mysql_use_result()). This works only for forward-only cursors. This option is very important in dealing with large tables when you don't want the driver to cache the entire result set.
2097152	Force the use of Forward-only cursor type. In case of applications setting the default static/dynamic cursor type, and one wants driver to use non-cache result sets, then this option will ensure the forward-only cursor behavior.

SQL Script File

Output a SQL Script file that can be executed on the target database server. Several predefined SQL-dialects are available, to define your own use the Connect String tags and the SQL Settings.

Connect String:
FILE;Type=SQL; Selects SQL Script File
TextDataType=VARCHAR(255); Sets Text data type
MemoDataType=TEXT; Sets Mext data type
DoubleDataType=FLOAT; Sets Double data type
LongDataType=INTEGER; Sets Long data type
DateDataType=DATETIME; Sets Date data type

SQL Server and MSDE

Connect String:
ODBC;DRIVER={SQL Server}; Selects SQL Server ODBC Driver
Database=mydb; Database Name
Server=(local); Server
Trusted_Connection=", "yes; Trusted Connection: yes = Login with Windows username ,no = use Uid and Pwd
Uid=Username; Userame is required when not using a Trusted Connection
Pwd=Password; Password is required when not using a Trusted Connection

XML File

The format of the connect string for XML files is:

File; Type=XML;
RootTag=<roottag>;
RecordTag=<recordtag>;
DateFormat=<dateformat>;
NumericFormat=<numericformat>;

Limitations: Append is not supported with XML files.

Tags:

Type=XML;
Specifies XML output.

RootTag=<roottag>;
The name of the root element of the XML file. Default is “Table”.

RecordTag=<recordtag>;
The name of the record elements of the XML file. Default is “Record”.

DateFormat=<dateformat>;
Set formatting for dates. See below for detailed specification.

NumericFormat=<numericformat>;
Set formatting for numbers. See below for detailed specification.

Other Databases (ODBC)

To import into MS-SQL Server or other ODBC databases, click the ODBC button. Then select the data source for the import, or click the New button to create a new data source. Restrictions may apply depending on the ODBC driver used. Specifically, for some databases the table viewer/browser might not work, although importing into the table will work. In this case use a table viewer that came with the database.

Date and Numeric Format Strings

Date and Numeric format strings are used in the diverse Connect Strings for file output, and in the SQL Date Format property for ODBC database connections. The default date format for file connect strings is "\#mm/dd/yyyy\#"

Date/Time Format Strings

Character	Description
(\)	Display the next character in the format string. To display a character that has special meaning as a literal character, precede it with a backslash (\). The backslash itself isn't displayed. Using a backslash is the same as enclosing the next character in double quotation marks. To display a backslash, use two backslashes (\\). Examples of characters that can't be displayed as literal characters are the date-formatting and time-formatting characters (a, c, d, h, m, n, p, q, s, t, w, y, / and :), the numeric-formatting characters (#, 0, %, E, e, comma, and period), and the string-formatting characters (@, &, <, >, and !).
(:)	Time separator. In some locales, other characters may be used to represent the time separator. The time separator separates hours, minutes, and seconds when time values are formatted. The actual character used as the time separator in formatted output is determined by your system settings.
(/)	Date separator. In some locales, other characters may be used to represent the date separator. The date separator separates the day, month, and year when date values are formatted. The actual character used as the date separator in formatted output is determined by your system settings.
c	Display the date as ddddd and display the time as ttttt, in that order. Display only date information if there is no fractional part to the date serial number; display only time information if there is no integer portion.
d	Display the day as a number without a leading zero (1 – 31).
dd	Display the day as a number with a leading zero (01 – 31).
ddd	Display the day as an abbreviation (Sun – Sat).
dddd	Display the day as a full name (Sunday – Saturday).
ddddd	Display the date as a complete date (including day, month, and year), formatted according to your system's short date format setting. For Microsoft Windows, the default short date format is m/d/yy.
dddddd	Display a date serial number as a complete date (including day, month, and year) formatted according to the long date setting recognized by your system. For Microsoft Windows, the default long date format is mmmm dd, yyyy.
w	Display the day of the week as a number (1 for Sunday through 7 for Saturday).
ww	Display the week of the year as a number (1-54).

m	Display the month as a number without a leading zero (1-12). If m immediately follows h or hh, the minute rather than the month is displayed.
mm	Display the month as a number with a leading zero (01-12). If m immediately follows h or hh, the minute rather than the month is displayed.
mmm	Display the month as an abbreviation (Jan-Dec).
mmmm	Display the month as a full month name (January-December).
q	Display the quarter of the year as a number (1-4).
y	Display the day of the year as a number (1-366).
yy	Display the year as a 2-digit number (00-99).
yyyy	Display the year as a 4-digit number (100-9999).
h	Display the hour as a number without leading zeros (0-23).
hh	Display the hour as a number with leading zeros (00-23).
n	Display the minute as a number without leading zeros (0-59).
nn	Display the minute as a number with leading zeros (00-59).
s	Display the second as a number without leading zeros (0-59).
ss	Display the second as a number with leading zeros (00-59).
t t t t t	Display a time as a complete time (including hour, minute, and second), formatted using the time separator defined by the time format recognized by your system. A leading zero is displayed if the leading zero option is selected and the time is before 10:00 A.M. or P.M. For Microsoft Windows, the default time format is h:mm:ss.
AM/PM	Use the 12-hour clock and display an uppercase AM with any hour before noon; display an uppercase PM with any hour between noon and 11:59 P.M.
am/pm	Use the 12-hour clock and display a lowercase AM with any hour before noon; display a lowercase PM with any hour between noon and 11:59 P.M.
A/P	Use the 12-hour clock and display an uppercase A with any hour before noon; display an uppercase P with any hour between noon and 11:59 P.M.
a/p	Use the 12-hour clock and display a lowercase A with any hour before noon; display a lowercase P with any hour between noon and 11:59 P.M.
AMPM	Use the 12-hour clock and display the AM string literal as defined by your system with any hour before noon; display the PM string literal as defined by your system with any hour between noon and 11:59 P.M. AMPM can be either uppercase or lowercase, but the case of the string displayed matches the string as defined by your system settings. For Microsoft Windows, the default format is AM/PM.

Examples:

The following are examples of user-defined date and time formats for December 7, 1958 8:50:35 PM

Format	Display
m/d/yy	12/7/58
d-mmm	7-Dec

d-mmmm-yy	7-December-58
d mmmm	7 December
mmmm yy	December 58
hh:mm AM/PM	08:50 PM
h:mm:ss a/p	8:50:35 p
h:mm	20:50
h:mm:ss	20:50:35
m/d/yy h:mm	12/7/58 20:50
\mmm\ddd	m12d07

NumericFormat Tag

The following table identifies characters you can use to create user-defined numeric formats. The default date format is "", i.e. no formatting is done..

CharacterDescription

Character	Description
(0)	Digit placeholder. Display a digit or a zero. If the expression has a digit in the position where the 0 appears in the format string, display it; otherwise, display a zero in that position. If the number has fewer digits than there are zeros (on either side of the decimal) in the format expression, display leading or trailing zeros. If the number has more digits to the right of the decimal separator than there are zeros to the right of the decimal separator in the format expression, round the number to as many decimal places as there are zeros. If the number has more digits to the left of the decimal separator than there are zeros to the left of the decimal separator in the format expression, display the extra digits without modification.
(#)	Digit placeholder. Display a digit or nothing. If the expression has a digit in the position where the # appears in the format string, display it; otherwise, display nothing in that position. This symbol works like the 0 digit placeholder, except that leading and trailing zeros aren't displayed if the number has the same or fewer digits than there are # characters on either side of the decimal separator in the format expression.
(.)	Decimal placeholder. In some locales, a comma is used as the decimal separator. The decimal placeholder determines how many digits are displayed to the left and right of the decimal separator. If the format expression contains only number signs to the left of this symbol, numbers smaller than 1 begin with a decimal separator. To display a leading zero displayed with fractional numbers, use 0 as the first digit placeholder to the left of the decimal separator. The actual character used as a decimal placeholder in the formatted output depends on the Number Format recognized by your system.
(%)	Percentage placeholder. The expression is multiplied by 100. The percent character (%) is inserted in the position where it appears in the format string.

(,)	Thousand separator. In some locales, a period is used as a thousand separator. The thousand separator separates thousands from hundreds within a number that has four or more places to the left of the decimal separator. Standard use of the thousand separator is specified if the format contains a thousand separator surrounded by digit placeholders (0 or #). Two adjacent thousand separators or a thousand separator immediately to the left of the decimal separator (whether or not a decimal is specified) means "scale the number by dividing it by 1000, rounding as needed." For example, you can use the format string "##0,," to represent 100 million as 100. Numbers smaller than 1 million are displayed as 0. Two adjacent thousand separators in any position other than immediately to the left of the decimal separator are treated simply as specifying the use of a thousand separator. The actual character used as the thousand separator in the formatted output depends on the Number Format recognized by your system.
(:)	Time separator. In some locales, other characters may be used to represent the time separator. The time separator separates hours, minutes, and seconds when time values are formatted. The actual character used as the time separator in formatted output is determined by your system settings.
(/)	Date separator. In some locales, other characters may be used to represent the date separator. The date separator separates the day, month, and year when date values are formatted. The actual character used as the date separator in formatted output is determined by your system settings.
(E- E+ e- e+)	Scientific format. If the format expression contains at least one digit placeholder (0 or #) to the right of E-, E+, e-, or e+, the number is displayed in scientific format and E or e is inserted between the number and its exponent. The number of digit placeholders to the right determines the number of digits in the exponent. Use E- or e- to place a minus sign next to negative exponents. Use E+ or e+ to place a minus sign next to negative exponents and a plus sign next to positive exponents.
- + $ ( )	Display a literal character. To display a character other than one of those listed, precede it with a backslash (\) or enclose it in double quotation marks (" ").
(\)	Display the next character in the format string. To display a character that has special meaning as a literal character, precede it with a backslash (\). The backslash itself isn't displayed. Using a backslash is the same as enclosing the next character in double quotation marks. To display a backslash, use two backslashes (\\). Examples of characters that can't be displayed as literal characters are the date-formatting and time-formatting characters (a, c, d, h, m, n, p, q, s, t, w, y, / and :), the numeric-formatting characters (#, 0, %, E, e, comma, and period), and the string-formatting characters (@, &, <, >, and !).
("ABC")	Display the string inside the double quotation marks (" "). To include a string in format from within code, you must use Chr(34) to enclose the text (34 is the character code for a quotation mark (")).

Command Line Options (Pro Version Only)

It is possible to run the Import Wizard from a command line, the Run option in the Start menu or from within another application or batch file.

impwiz.exe <modelfile> [/O] [/L <logfile>] [/S <sourcefile>] [/C <connect>] [/D <dbfile>] [/T <table>] [/A <action>] [/W <+|->]

Text between [...] is optional, text between <...> needs to be replaced with the appropriate argument. The switches are not case sensitive and a space between a switch and the following argument is optional. The syntax can be shown with impwiz.exe /?

Available switches:

/O Open <Modelfile>, do not import
/L Specify logfile
/S Specify sourcefile
/C Specify destination database connect string
/D Specify destination database filename
/T Specify destination table name
/A Import Action: sets the import action(s) to perform.
1: Create new table, remove existing table and recreate it
2: Empty table, clear all existing records from the table before importing.
4: Insert, add imported records to the end of the table.
8: Update, replace existing records with imported records when the update key fields match
To add the numbers to combine actions, for example the value 5 (=1+4) will create a new table and then insert the records. If neither 1 or 2 is specified the records will be appended to an existing table.
/W Show Warnings + yes or – no

If an optional argument is omitted then values as defined in the model are used.

Return values:

0 Import completed without warnings or errors.
1 Import completed with warnings.
2 Import failed.

Examples

impwiz /? Shows synopsis

impwiz somemodel /o opens modelfile "somemodel.iwm" but does not run it.

impwiz mdl /o /sc:\import*.* opens modelfile "mdl.iwm" with source files "c:\import*.*" but does not run the model.

impwiz c:\models\model.mdl performs an import as defined in model file "c:\models\model.mdl"

impwiz test /s myfile.txt /a8 /t mytable runs model "test.iwm" using update mode and destination table "mytable"

Batch File

The return value can be used together with the DOS batch file command ERRORLEVEL for error handling. For example, the following batch file runs an import and prints out "Import OK", "Import Warnings", or "Import Failed" based on the result of the import. This batch file is available in as "iwdemo.bat" in the Samples directory.

```
impwiz "sample1 fixed.iwm"
IF ERRORLEVEL 2 GOTO fail
IF ERRORLEVEL 1 GOTO warn
echo Import OK
GOTO end
:warn
echo Import Warnings
GOTO end
:fail
echo Import Failed
:end
```

International Versions

The Import Wizard user interface is multilingual. All text elements are stored in language files. The language files have to be placed in the same directory as the main DLL file (iw8dll.dll), and have to be named "langXX.dat". (Where XX stands for a two character language code.)

Language files are optional, the default language (US-English) is build into the DLL itself. If a certain text element is not available in a language file, then the default US-English version of that text element will be displayed. The language files itself are plain text files that can be edited with any text editor.

The "_langUS.dat" file can be used as a template for your own language file. The file is installed with the Import Wizard distribution in the application directory, and contains a copy of the build-in US-English language file. Please see this file for further details.

Language files are not maintained by us, they are created by Import Wizard users. If you have created a new language file, or if you have corrected an existing language file, we would greatly appreciate it to receive a copy for publication with the next release of Import Wizard.

Copyrights, Warranty, License Agreement

You should carefully read the following terms and conditions before using the software.

Copyrights

Import Wizard version 8, including the Import Wizard Pro, Import Wizard Access Add-In, and Import Wizard Excel Add-In, (the "Software") is owned by Beside Company Inc. and is protected by copyright laws and international treaties.

LICENSE AGREEMENT

Use of Unregistered Shareware Version

You are hereby licensed to:

- Use the unregistered shareware version of the Software for a 30 day evaluation period,
- Make as many copies of the unregistered shareware version of the Software as you wish,
- Give exact copies of the unregistered shareware version of the Software to anyone, and
- Distribute the shareware version in its unmodified form via electronic or other means.

You are specifically prohibited from charging, or requesting donations, for any such copies, however made.

Evaluation and Registration

This is not free Software. You may not use this software beyond the initial 30 day evaluation period unless you register the Software. See Ordering Software, Registration

Use of Registered Version

Registration of the Software grants you a non-exclusive right to load and use the Software. The Software is "loaded" onto a computer when it is either copied to the permanent memory of the computer (the hard disk) or loaded into the temporary memory (RAM), such as when a computer runs the software from a network server, or from removable disk media.

Single User License

You may use the Software only on a single computer, or on up to two computers used exclusively by the same individual user.

Multi User License

The Software can be loaded in such a way that the number of users is less than or equal to the number of licenses owned. A user is any person that has access to the licensed software, regardless whether this person is actually using the software.

Site License

If a site license is purchased, the Software may only be used at the business address (site) for which the software was bought, regardless of the number of users or computers at that address.

Developer License

Registration of the Software grants you a non-exclusive right to embed the Import Wizard DLL (iw60dll.dll) into your own custom software application. You are allowed to distribute your application together with the Import Wizard DLL and the software registration key without royalty fees if your application meets the following criterion: ***Your application provides functionality that does not depend on the Import Wizard DLL.*** (In other words, your application does more than importing data.) If, to the judgement of Beside Company Inc. this criterion is not met, Beside Company Inc. reserves the right to revoke the royalty free distribution right. You are specifically prohibited from distributing (in modified or unmodified form) the sample application(s) contained in the Developer Kit with a registration key.

Other Restrictions

You may not transfer your rights under this agreement. You must not rent or lease this Software. You may not decompile or disassemble the Software. You may not distribute the registered version of the Software.

DISCLAIMER OF WARRANTY

THIS SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" AND WITHOUT WARRANTIES AS TO PERFORMANCE OF MERCHANTABILITY OR ANY OTHER

WARRANTIES WHETHER EXPRESSED OR IMPLIED. BECAUSE OF THE VARIOUS HARDWARE AND SOFTWARE ENVIRONMENTS INTO WHICH THIS PROGRAM MAY BE PUT, NO WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE IS OFFERED. GOOD DATA PROCESSING PROCEDURE DICTATES THAT ANY PROGRAM BE THOROUGHLY TESTED WITH NON-CRITICAL DATA BEFORE RELYING ON IT. ALSO, PRIOR TO THE USE OF THIS SOFTWARE WITH ANY DATA, A COMPLETE BACKUP OF THE DATA SHOULD BE MADE (AND RETAINED UNTIL IT IS DETERMINED THAT THE MODIFICATIONS MADE BY THIS SOFTWARE ARE CORRECT AND HAVE HAD NO UNDESIRED EFFECTS). THE USER MUST ASSUME THE ENTIRE RISK OF USING THE PROGRAM. ANY LIABILITY OF THE SELLER WILL BE LIMITED EXCLUSIVELY TO PRODUCT REPLACEMENT OR REFUND OF PURCHASE PRICE.